IN THE CLAIMS:

Please find below a listing of all of the pending claims. The statuses of the claims are set forth in parentheses.

1. (Currently Amended) A method for accessing memory in a multiprocessor system, said multiprocessor system including processors and a memory, said method comprising:

storing data in one or more of a plurality of the processors and the memory;

in one of the one or more of said plurality of processors ~~and said memory~~ where the data is stored, modifying the data;

in the one of the one or more of said plurality of processors ~~and said memory~~ that modified the data, associating the modified data with state information indicating that the modified data is valid;

in the others of the plurality of processors and said memory where the data is stored, associating the data with state information indicating that the data is invalid;

from a requesting processor, issuing a request for the modified data to one or more other processors and memory;

in each of the processors and memory that receive the request, checking to determine whether a stored copy of the data is valid or invalid;

in the processor ~~or memory~~ having the valid copy of the data, responding to the request;

in the processors ~~or~~ and memory that have invalid copies of the data, dropping the request without responding to the request; and

in the processor ~~or memory~~ having the valid copy of the data, returning the valid copy of the requested data to the requesting processor.

2. (Original) The method of claim 1 in which:

each of the processors communicates with the memory via a memory controller and each of the processors has a point-to-point link with the memory controller for issuing a request for a block of data to the memory controller.

3. (Original) The method of claim 2 in which:

each point-to-point link includes two dedicated and unidirectional links.

4. (Original) The method of claim 2 in which the point-to-point links are control links for sending and receiving requests for blocks of data.

5. (Original) The method of claim 2 in which each of the processors has a control path point-to-point link for sending and receiving requests for blocks of data, and a data path point-to-point link for sending and receiving blocks of data.

6. (Canceled).

7. (Original) The method of claim 1 including:

tracking an identification of a processor that currently has a data block; and

in response to a cache miss in a requesting processor, using the identification to specifically target a read request to the processor that currently has the requested data block.

8. (Previously Presented) The method of claim 1 including:

maintaining a directory indicating the one or more processors that have a copy of the data;

using the directory to issue a write invalidation or write update only to the processors that have the copy of the data.

9. (Currently Amended) A multiprocessor system comprising:

two or more processors, each in communication with a shared memory via a memory controller;

said two or more processors and said memory being operable to store and modify data;

when one of said two or more processors ~~and said memory~~ modifies the stored data, the one of said two or more processors ~~and said memory~~ that modified the data being operable to associate the modified data with state information indicating that the modified data is valid, and the others of said two or more processors and said memory that did not modify the stored data being operable to associate the modified data with state information indicating that the stored data is invalid;

each of the two or more processors being in communication with the memory controller for issuing a request for the modified data to the others of the two or more processors and said memory;

when one of the two or more processors issues a request for the modified data, each of the two or more processors and the shared memory that receives the request being operable to check itself to determine whether a stored copy of the data is valid or invalid;

wherein the one of said two or more processors ~~and said memory~~ that modified the data is configured to respond to the request, and the processors ~~or~~ and memory having invalid copies of the data are configured to drop the request without responding to the request; and

wherein the one of said two or more processors ~~and said memory~~ that modified the data is configured to return the valid copy of the modified data to the one of the two or more processors or memory that issued the request.

10. (Original) The system of claim 9 in which:

each of the processors communicates with the memory via a memory controller and each of the processors has a point-to-point link with the memory controller for issuing a request for a block of data to the memory controller.

11. (Original) The system of claim 10 in which:

each point-to-point link includes two dedicated and unidirectional links.

12. (Original) The system of claim 10 in which the point-to-point links are control links for sending and receiving requests for blocks of data.

13. (Original) The system of claim 10 in which each of the processors has a control path point-to-point link for sending and receiving requests for blocks of data, and a data path point-to-point link for sending and receiving blocks of data.

14. (Previously Presented) The system of claim 9 including:

a directory indicating which processors have a copy of the data block;

wherein the processors are in communication with the directory to identify which other processors have a copy of the data block, and wherein the processors are configured to direct requests for the data only to processors that have a copy of the data.

15. (Original) The system of claim 14 wherein the directory is incorporated into the data block.

16. (Original) The system of claim 14 wherein the directory is stored in a separate memory that filters a request and forwards the request only to a processor or processors that have a copy of the data block.

17. (Original) The system of claim 14 wherein the memory controller is in communication with a shared cache, separate from caches of the processors, for buffering most frequently accessed data blocks.

18. (Original) The system of claim 9 wherein each block has state information indicating which processor currently has a valid copy of a data block, and wherein the processors utilize the state information to specially address a processor having the valid copy in response to a cache miss in a requesting processor.

19. (Currently Amended) A multiprocessor system comprising:

two or more processors, each in communication with a shared memory;

said two or more processors and said memory being operable to store and modify data;

when one of said two or more processors ~~and said memory~~ modifies the stored data, the one of said two or more processors ~~and said memory~~ that modified the data being operable to associate the modified data with state information indicating that the modified data is valid, and the others of said two or more processors and said memory that did not modify the stored data being operable to associate the modified data with state information indicating that the stored data is invalid;

each of the two or more processors being ~~in communication with the memory controller for issuing~~ operable to issue a request for the modified data to the others of the two or more processors and said memory;

when one of the two or more processors issues a request for the modified data, each of the two or more processors and the memory that receives the request being operable to check itself to determine whether a stored copy of the data is valid or invalid;

wherein the one of said two or more processors ~~and said memory~~ that modified the data is configured to respond to the request, and the processors ~~or~~ memory having invalid copies of the data are configured to drop the request without responding to the request; and

wherein the one of said two or more processors ~~and said memory~~ that modified the data is configured to return the valid copy of the modified data to the one of the two or more processors or memory that issued the request.

20. (Original) The system of claim 19 wherein each of the processors and the shared memory is in communication with a control path interconnect, and each of the processors is in communication with the control path interconnect via a point-to-point link for receiving and sending requests for blocks of data;

each of the processors having a corresponding request queue connecting the point-to-point link of the processor to the control path interconnect, and each of the processors having a corresponding snoop queue connecting the point-to-point link of the processor to the control path interconnect;

the request queue in communication with a corresponding processor for buffering requests for blocks of data by the processor and issuing the requests to other processors via the control path interconnect; and

the snoop queue in communication with a corresponding processor for buffering requests for blocks of data destined for the processor.

21. (Previously presented) The method of claim 1, wherein the step of returning the valid copy of the requested data comprises returning the valid copy asynchronously.

22. (Previously presented) The system of claim 9, wherein the processor or the shared memory responding to the request is configured to respond to the request asynchronously.

23. (Previously presented) The system of claim 19, wherein the processor or the shared memory responding to the request is configured to respond to the request asynchronously.